Team 1 Project Backlog: WeChef

Team Members:
Sean Chew (schewshu@purdue.edu)
Zuyuan Fan (fan164@purdue.edu)
Yuting Guo (guo312@purdue.edu)
Qi Meng (meng46@purdue.edu)
Ling Zhang (zhan2275@purdue.edu)

Problem Statement

Have you ever found yourself standing in the kitchen with a handful of ingredients that you have no idea what to do with? Oftentimes, just finding the correct recipes can be a hassle. Our team is envisioning a convenient and user-friendly mobile app called WeChef as the solution to this problem.

The mobile app will provide individuals with an easy way to discover new recipes as well as find recipes based on the current available ingredients at hand. In addition, WeChef will allow individuals to live stream their cooking sessions for the general public to view and learn from them. They will also be able to upload photos/videos for individuals who prefer to learn at their own pace.

What makes us stand out from our competitors, such as Tasty, is that our app is user-based, whereby anyone can upload their recipes. Tasty, on the other hand, is fully controlled by the professional editors, and users can only view the recipes posted by them.

In conclusion, our app aims to provide a platform for anyone who aspires to improve their cooking skills and we hope to grow a fun and friendly community for cooking enthusiasts.

Background Information

- Domain

  The domain of our application is food and social networking since users could see, add to favorites, and comment on cooking tutorials created by other users.

- Targeted users

  Who doesn't appreciate a delicious home-cooked meal? While there are many people who have great cooking skills, there are also a lot of people who are still learning or wanting to learn how to cook. Our targeted users are the people who are interested in learning how to cook and explore new recipes, as well as individuals who are passionate about cooking and want to share their recipes by teaching the rest of the culinary community.

- Similar platforms

  There are a couple of cooking apps currently available in the App Store, including Tasty and Food Network In the Kitchen. They provide similar functionalities such as step-by-step recipe instructions, video tutorials, and meal recommendations for various occasions.

- Limitations of other solutions

  Although the existing apps have proven to be useful, they are fully controlled by professional editors and do not allow for common users to post and upload recipes online. Our app is mainly user-based and hopes to provide a platform for cooking enthusiasts to come together and share their different recipes within the culinary community.

- Environment

  We plan to develop an iOS application using React Native framework and Atom as our IDE. We will use MongoDB as our database to store user information. We will use Git as our version control system.

Functional Requirements

1. As a user, I want to create an account
2. As a user, I want to sign in to the application
3. As a user, I want to sign out of the application
4. As a user, I want to set a profile name
5. As a user, I want to set a profile picture
6. As a user, I want to update my profile name
7. As a user, I want to update my profile picture
8. As a user, I want to search for a particular recipe from the search bar using the recipe name
9. As a user, I want to search for a recipe using particular ingredients
10. As a user, I want to see ingredients and the amount used in the recipe
11. As a user, I want to see step-by-step details of the recipes
12. As a user, I want to see pictures of the dish posted by the author of the recipe
13. As a user, I want to watch cooking videos posted by the author of the recipe
14. As a user, I want to see pictures of dishes cooked by other users who used the recipe
15. As a user, I want to see comments and questions posted by other users who used the same recipe
16. As a user, I want to add ingredients of a recipe to my shopping list
17. As a developer, I want to provide the users unit conversion function
18. [If time permits] As a user, I want to view my shopping list by recipe
19. [If time permits] As a user, I want to combine recipe ingredients in shopping list
20. [If time permits] As a user, I want to remove all ingredients from my shopping list
21. [If time permits] As a user, I want to remove ingredients in a recipe from my shopping list
22. [If time permits] As a user, I want to see the homepage of the author of a particular dish
23. [If time permits] As a user, I want to add a particular recipe to favorites
24. [If time permits] As a user, I want to like a particular recipe

25. As a user, I want to post detailed steps of my own recipes
26. As a user, I want to post pictures of my own recipes
27. [If time permits] As a user, I want to post videos of my own recipes
28. As a user, I want to ask questions about a particular recipe
29. As a user, I want to post comments under the recipes that I have used
30. As a user, I want to post pictures of my version of the dishes under the recipes that I have used
31. As a user, I want to rate subjective quality of the recipes that I have used
32. As a user, I want to rate subjective difficulty of the recipes that I have used
33. As a user, I want to edit the recipe that I have posted
34. [If time permits] As a user, I want to follow the users whom I am interested in

Non-Functional Requirements

- Architecture
  We've decided to develop this application on iOS since it is one of the most popular mobile operating systems, and so we could utilize this large platform to promote this application.

  We will separate our product to data layer, logic layer, and presentation layer. Data layer consists of our database and object storage. Logic layer connects and communicate between data layer and presentation layer to respond to requests from our frontend client. Client application interface serves as the presentation layer of our product; it displays information to end users and allow them to use the core functions of the application.

  By separating the product into the layers, we eliminate dependency among function components and gain more flexibility on choices of framework and tools. This also ensures scalability of server, database, and object storage in an on-demand manner.

- Framework and Tools

  We will construct our API in Node.js environment with Express framework to take care of basic HTTP infrastructures, like routing and handling requests. The API route can be triggered by the client application and perform backend logic and database operation according to the requests. The API will be hosted on Heroku for its convenience in deployment and management. We will store information of users and recipes in MongoDB served on mLab added on Heroku. Photos will be stored with another Heroku add on, Cloudinary.

  For the client side, we will be using React Native as a front-end development framework. Facebook will be used for user authentication proposes to make the sign up/login process more convenient and secure.

- Security

  As our application will be built on matured platforms like Node JS Express, MongoDB, etc, our application will be equipped with security features that are provided by these services. For example, MongoDB provides various security features that include authentication, access control, encryption, etc. In addition, we will extensively use APIs, such as Facebook Login. Facebook Login provides features, such as access tokens and permissions, to make it safe and secure for applications to use.

- Usability

  The success of our app is ultimately determined by our users. There, usability and user experience is of utmost importance to us and will be a driven factor of our UI/UX design process. We try to make the interface as user-friendly as possible, eliminating as much hassle as we can along the way. Starting from the login and sign-up process, we only allow for third-party authentication login (such as Facebook login) to free our users from memorizing yet another username and password. This also eliminates the initial complicated sign-up process and allows users to dive right in and enjoy the exciting services right away. Once logged in, the user-friendly interface allows our users to quickly familiarize themselves with the app to utilize the full-range of functionalities. In addition, since our application will involve extensive use of pictures and possibly videos, we want to provide a smooth picture and video loading process by integrating lazy loading feature.

- Scalability
  Given that our original thought is that our product will be used by our group members and some of our friends, there may not be many problems with scalability. If our product were to be released to a wider group of users, we can easily upgrade Heroku Dynos and increase the Cloudinary and mLab storage in order to sustain a relatively high speed and keep the user data.

Github Repository

https://github.com/taeroro/WeChef